**Resumen**

Una de las aplicaciones más frecuentes, en programación es la ordenación.

• Existen dos técnicas de ordenación fundamentales en gestión de datos: ordenación de listas y ordenación de archivos.

• Los datos se pueden ordenar en orden ascendente o en orden descendente.

• Cada recorrido de los datos durante el proceso de ordenación se conoce como pasada o iteración.

• Los algoritmos de ordenación básicos son:

• Selección.

• Inserción.

• Burbuja.

• Los algoritmos de ordenación más avanzados son:

• Shell.

• Mergesort.

• Quicksort.

• La eficiencia de los algoritmos de burbuja, inserción y selección es 0(n2).

• La eficiencia de los algoritmos heapsort, radixsort, mergesort y quicksort es 0(n Log n).

• La búsqueda es el proceso de encontrar la posición de un elemento destino dentro de una lista. • Existen dos métodos básicos de búsqueda en arrays: búsqueda secuencial y binaria.

• La búsqueda secuencial se utiliza normalmente cuando el array no está ordenado. Comienza en el principio del array y busca hasta que se encuentra el dato buscado y se llega al final de la lista.

• Si un array está ordenado, se puede utilizar un algoritmo más eficiente denominado búsqueda binaria.

• La eficiencia de una búsqueda secuencial es 0(n).

• La eficiencia de una búsqueda binaria es 0(log n).

**Taller**

Por: Andrés Restrepo, Camilo Arango e Isabella Navarro

1. (10)Se desea eliminar todos los números duplicados de una lista

Por ejemplo

Si se le ingresan los valores [4,7,11,4,9,5,11,7,3,5]

Se debe cambia a [4,7,11,9,5,3]

1. (10)Elimine los elementos duplicados de un vector ordenado.

¿Cuál es la eficiencia del método?

Text

Description automatically generated

Se puede ver que con el vector ordenado el algoritmo corre mucho más rápido a comparación del vector desordenado.

1. (5)Dada la siguiente lista

[47,3,21,32.56,92]

Después de 2 “pasadas” de un algoritmo de ordenación, la lista ha quedado dispuesto asi

[3,21,47,32,56,92]

¿Qué algoritmo de ordenación se esta utilizando (selección, burbuja o inserción)?

Justifique su respuesta

Text

Description automatically generated Se utiliza el método de la burbuja porque como vemos, en la segunda pasada queda con el mismo vector que se dio en el enunciado del punto.

1. (10) Utilizar el algoritmo de ordenación SHELL, encuentre las pasadas e intercambios que se realizan para la ordenación de la siguiente lista



Text

Description automatically generated

Acá podemos evidenciar las pasadas e intercambios que se realizan en cada pasada.

1. (10) Dada una lista voto[0.......n-1], donde cada elemento de lista representa un voto en las elecciones. Suponga que cada voto se da como un número entero que representa el ID del candidato elegido. Desarrolle un algoritmo para determinar quién gana la elección. **Determine la complejidad del algoritmo**

La complejidad del algoritmo es n.

1. (10) Se cuenta con una lista de tuplas

futbolistasTup = [(1, "Casillas"), (15, "Ramos"), (3, "Pique"), (5, "Puyol"), (11, "Capdevila"), (14, "Xabi Alonso"), (16, "Busquets"), (8, "Xavi Hernandez"), (18, "Pedrito"), (6, "Iniesta"), (7, "Villa")]

si se aplica futbolistasTup.sort(key=lambda futbolista: futbolista[0])

1. Que resultado se obtiene al aplicar el método .sort

* Se organiza la lista de menor a mayor según el número de los jugadores

1. Que se esta especificando en los parámetro (key=lambda futbolista: futbolista[0])

* Que para ordenar la lista se toma el primer valor de cada elemento (jugadores) para ordenarlos de menor a mayor según su número.

1. Aplique este método a las listas de los punto 1,3, 4. Que conclusión puede obtener

* Ya que en los puntos 1,3 y 4 son solamente listas ordinarias/clásicas, no funciona este método para ellas.

1. Por favor según opinión realice una tupla con los mejores inventos del 2019 .

Donde usted califica el que mas le gusta o le parece importante.

**Anotación** la escala con la que usted cuenta es de 1 a 100 ( no tiene que asignar ninguno de los extremos si no lo desea)

* inventosTup = [(71, “El bastón inteligente”), (50, “Cirugía virtual”), (15, “AirPods Pro”), (30, “Pantallas de 8K”), (70, “La prótesis de mano robot”), (60, “Entregas a domicilio automáticas”), (1, “El basurero inteligente”)]

inventosTup.sort(key=lambda inventos: inventos[0])

1. (10) Diseñe e implemente una función para encontrar todos los valores negativos dentro de una lista dada. Tu función debería devolver una nueva lista que contiene los valores negativos.

¿Cuándo ocurre el peor de los casos y cuál es el tiempo de ejecución para ese caso?

Text

Description automatically generated

El peor de los casos ocurre cuando en la lista todos son negativos. En este caso, el tiempo de ejecución es 0.214 segundos.

1. (5) Dada la siguiente lista de números:

[21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]

¿Cuál sería el resultado a la lista que después de 3 llamadas recursivas de ordenamiento por mezcla?

¿Por qué?

Text

Description automatically generated

Después de la tercera llamada recursiva quedamos con [21] ya que el merge sort funciona dividiendo las listas en mitades hasta llegar a un elemento y comienza a comparar los elementos para ir ordenando de menor a mayor.

1. (5) **Dado el siguiente código evalué su complejidad.**

Su complejidad es log n.

**Determine que hace este esté algoritmo**

Este algoritmo lo que hace es crear un Set, que contiene elementos numéricos únicos en orden de menor a mayor, y los almacena dependiendo de su tamaño, para que de dichos elementos dentro del objeto Set, sigan estando almacenados en orden de menor a mayor.

**Explique cada uno de sus funciones y sus clases**

El método contructor crea una lista, donde se almacenar los datos.

El método \_\_len\_\_ retorna el tamaño de la lista.

El método \_\_contains\_\_ retorna si el elemento buscado se encuentra en la lista, llama al método de la clase findPosition para ver si dicho elemento existe.

El método add agrega un elemento único en la lista, y lo agrega correspondientemente a su tamaño.

El método remove primero busca si el elemento está en la lista, y luego lo saca.

El método isSubsetOf revisa que si los elementos de la lista pertenecen a otro objeto de la clase Set llamado setB.

El método \_\_iter\_\_ es un método de la clase que es el que nos permite recorrer objetos de tipo Set en un for.

El método findPosition es un método de búsqueda binaria, que recorre la lista ordenada con elementos únicos, y retorna la posición si el elemento es encontrado, de lo contrario retorna la posición donde iría el nuevo elemento a agregar.

st

1. (10) Construya un algoritmo para encontrar un valor específico en una matriz de valores ordenada por filas y columna.

El algoritmo toma como entrada una matriz de valores donde cada fila y cada columna están en orden, junto con un valor para ubicar en esa matriz. Devuelve si ese elemento existe en la matriz.

Por ejemplo, dado la siguiente matriz y buscar el 7, el algoritmo daría como resultado **sí**

Pero si se pide encontrar el número 0, el algoritmo daría como resultado **no**



1. (10) se tiene una lista A con 100 elementos A[ a1……a100 ]

B de 60 elementos B[ b1……b60 ]

Se desean resolver las siguientes tareas

1. Ordenar cada lista aplicando el método Quicksort
2. Crear una lista C que sea la unión de la lista A y B
3. Ordenar la lista C y visualizarla

Texto

Descripción generada automáticamente

1. (5) Investigue en que consisten las técnicas de ordenación Radixsort y Binsort, en lo posible haga un ejemplo.

El Binsort funciona de tal manera que se agrupan los números en ciertos buckets/casilleros y después se organizan dentro de los buckets para después unirlos y ordenar una lista de números.

Diagram

Description automatically generated with medium confidence

El Radixsort consiste en tomar los números de una lista y dependiendo del número en el que termine, los agrupa. Después toma el número inicial de cada número y los agrupa de menor a mayor, de la misma manera que los organiza al principio solo que con el primer digito del número en vez del último digito.

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Texto

Descripción generada automáticamente